Building and Training Spiking Neural Networks

William Gebhardt

Neural Adaptive Computing Laboratory

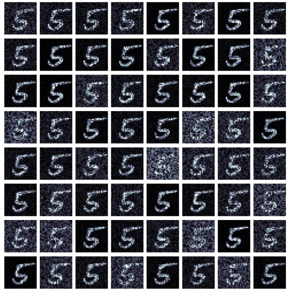
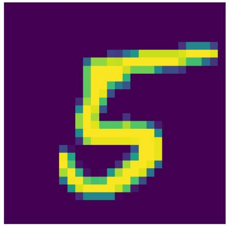
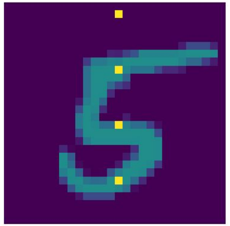


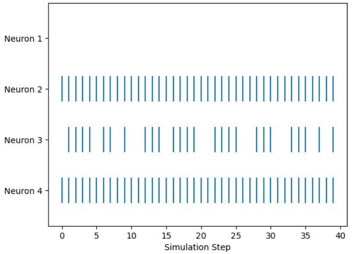
Image Reconstruction

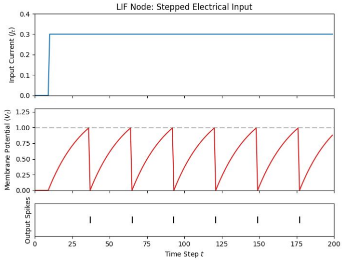
Input Image → Poisson Spike Train → Forward Passes → Reconstruction

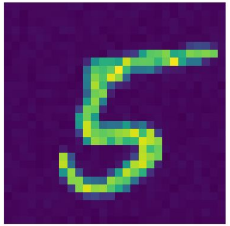




Pixels highlighted for visual purposes only







For this presentation we will be using instantaneous current

Coding up a layer of LIF neurons

- A vector of size N is needed to store the voltage of each neuron in the layer
- Hyperparameters
  - Leak: the rate at which voltage is lost
  - R: Affects the amount of voltage gained from the current input
  - dt: The time step between simulation steps
  - Tau: Affects the overall rate of change of the voltage

```
V = (V + (-V * leak + R * (s * W)) * (dt/tau))
```

Pulling out the spikes and depolarize

- Simple greater than check
- Reset the neurons the spiked to zero
- thr is the adaptive threshold

```
Sj = tf.cast(tf.math.greater(V, thr),dtype=tf.float32)
```

```
V = (V * (1.0 - Sj))
```

Adapting the threshold

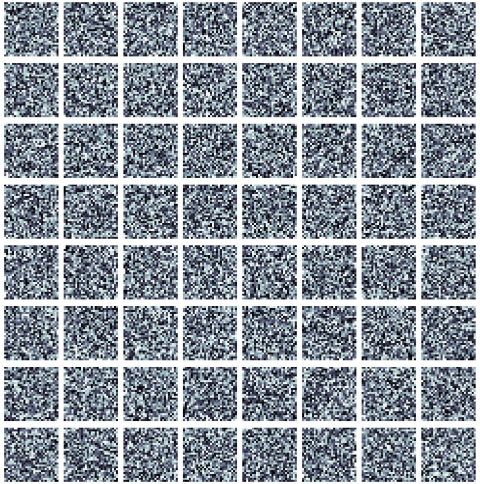
- ODE similar to voltage and current
- Alpha and beta are hyperparameters for adjusting the leak and gain

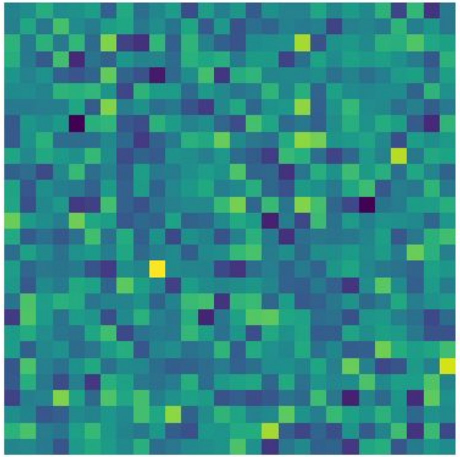
```
thr_leak = -thr * alpha
```

```
thr_gain = Sj * beta
```

```
thr.assign(thr + thr_leak + thr_gain)
```

What would happen if we run this?

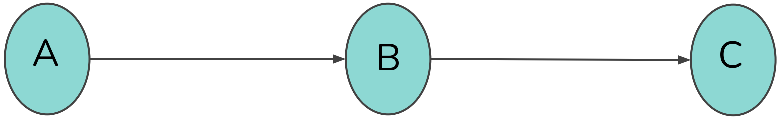




It needs to be trained

- Many methods of training
  - Backpropagation
  - Error Neurons
  - STDP
- Avoid backpropagation
  - Backprop through time
  - Spikes can not be differentiated

STDP



- **B** spikes
  - If **C** has spiked within a given window of time, this is a case of postsynaptic spiking and we decrease the weight between **B** and **C**

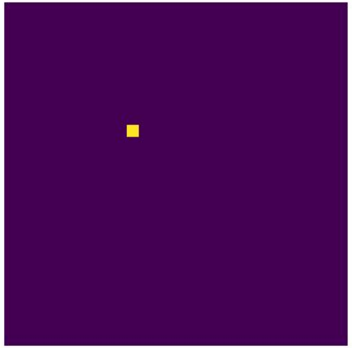
  - If **A** has spiked within a given window of time, this is a case of presynaptic spiking and we increase the weight between **A** and **B**

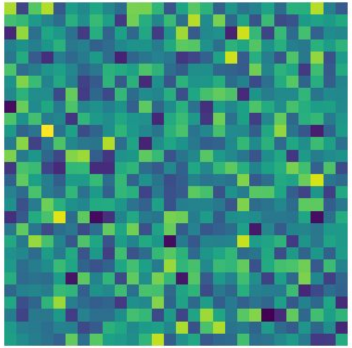
- **A** would only care about postsynaptic spiking whenever it spikes

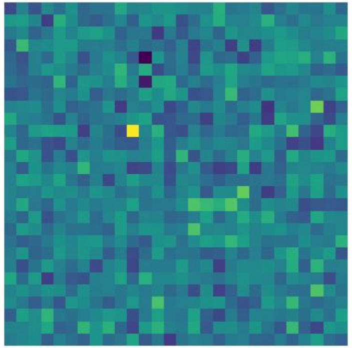
- **C** would only care about presynaptic spiking whenever it spikes

Problem: Blank Pixels

- Since no pixels other than the one spiked in the input layer no weights from the input layer to the output layer were updated
- Weight decay can combat this problem but this had scaling problems

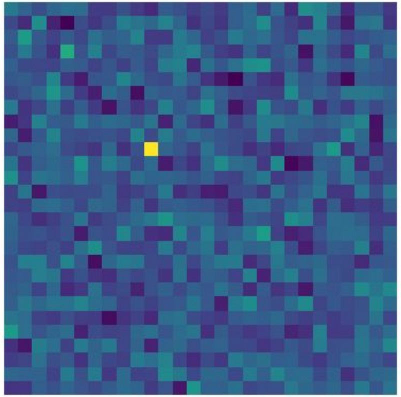




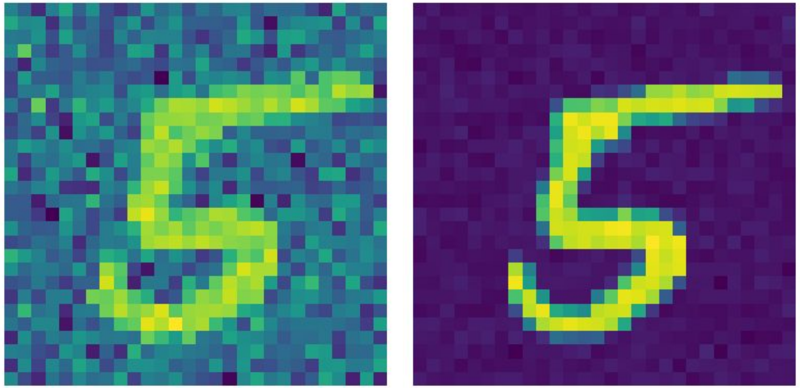


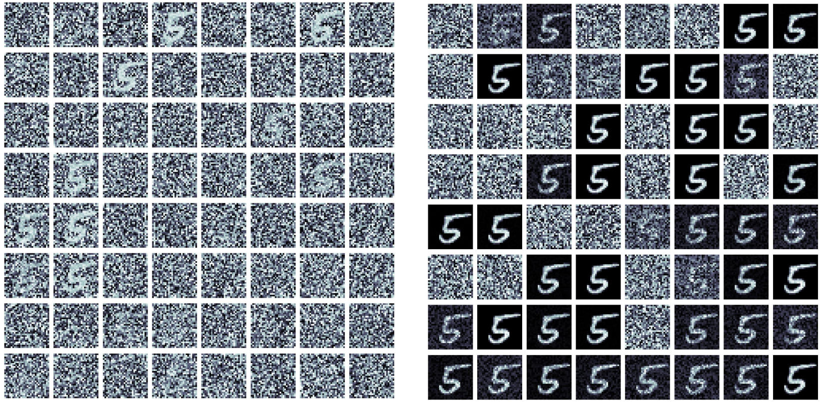
Postsynaptic Event based STDP

- Based on the method proposed by Amirhossein Tavanaei, Timothée Masquelier, Anthony Maida
    - Representation learning using event-based STDP
- Affects all incoming weights to a postsynaptic neuron when that neuron spikes
- Effectively has weight decay built in



Number Comparison





So how does it work

Scale the weights
```
delta = W * (1. + lmbda)
```

Extend presynaptic spikes
```
Mi = tf.zeros(W.shape) + tf.transpose(Si)
```

Calculate Update
```
dW = Mi * (1. - delta) + (1. - Mi) * (-delta)
```

Mask update
```
dW = dW * Sj * a
```

Assign update
```
W.assign(W + dW)
```

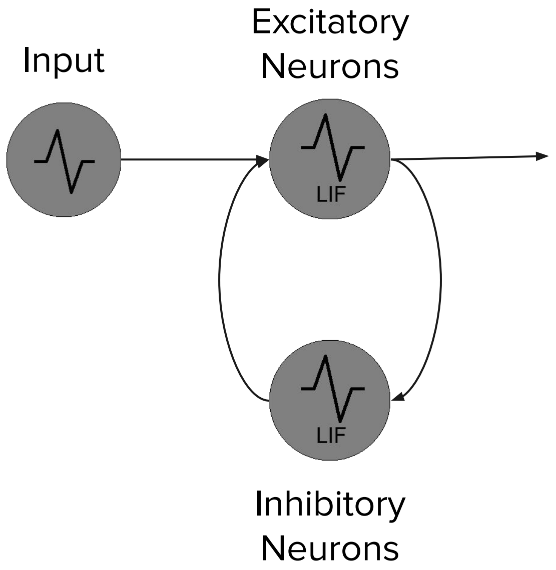
Are we done?

Other Needed Parts

- Inhibition
- Patching
- More Epochs

Inhibition

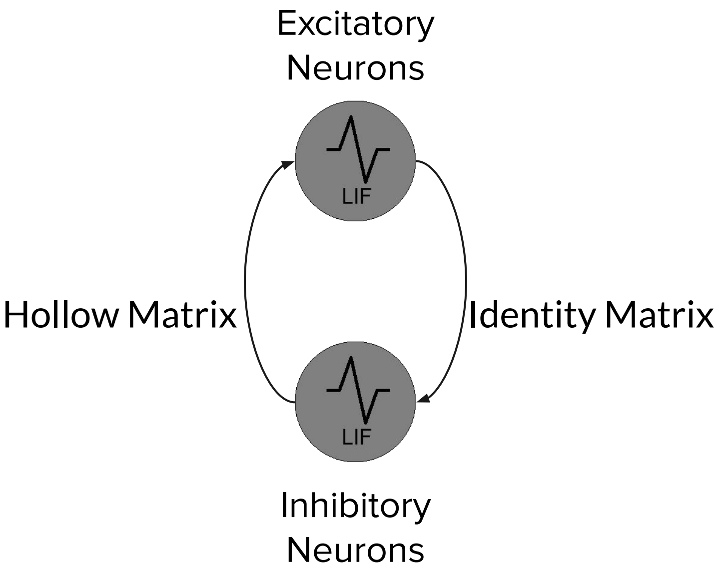
- Limit the number of neurons that are spiking together
- When an inhibitory neuron spikes it will remove voltage from all the excitatory neurons



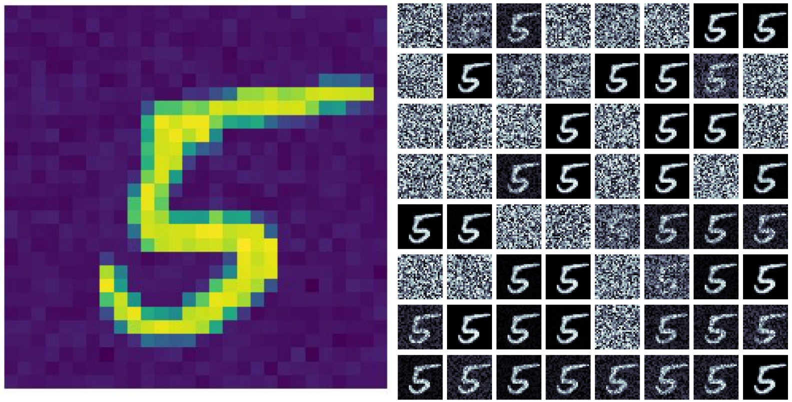
The Weight Setup

- Excitatory neurons are wired 1-to-1 with the identity matrix to inhibitory neurons

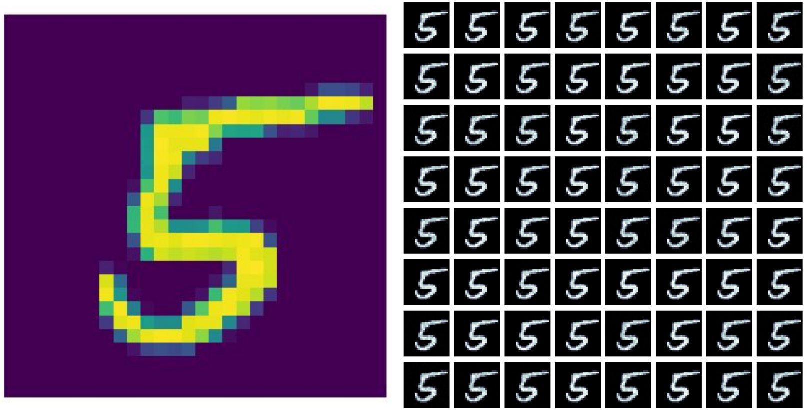
- Inhibitory neurons are wired 1-to-others with a hollow matrix back to excitatory neurons



Why it's needed

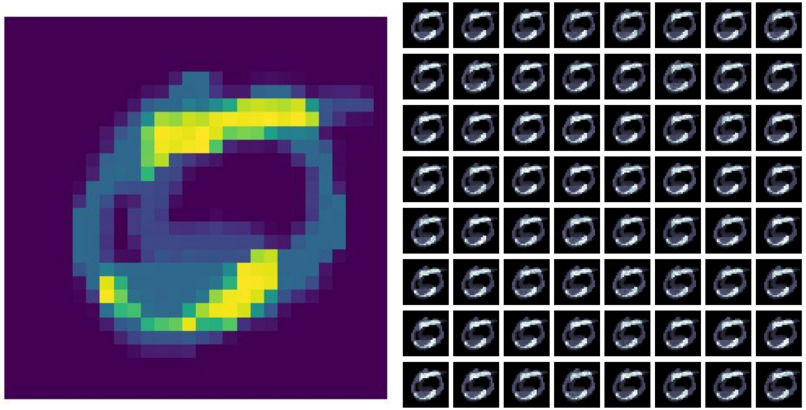


With inhibition



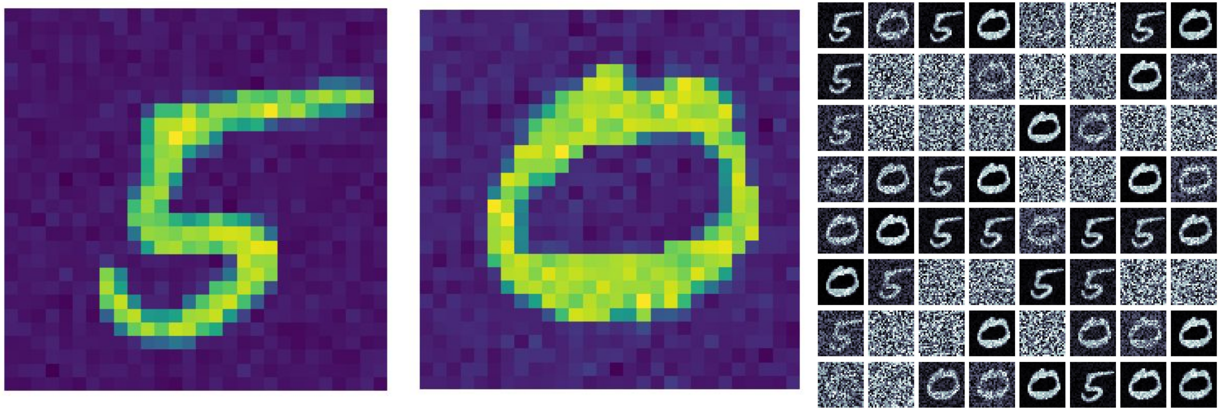
No inhibition

Where it breaks



No inhibition

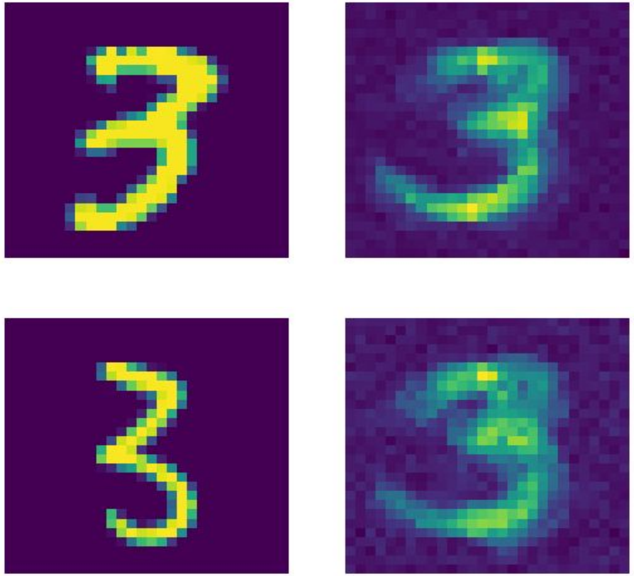
With inhibition

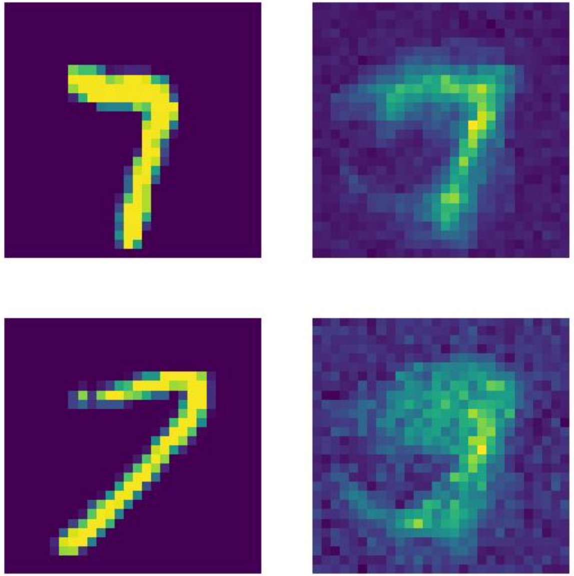


Patching

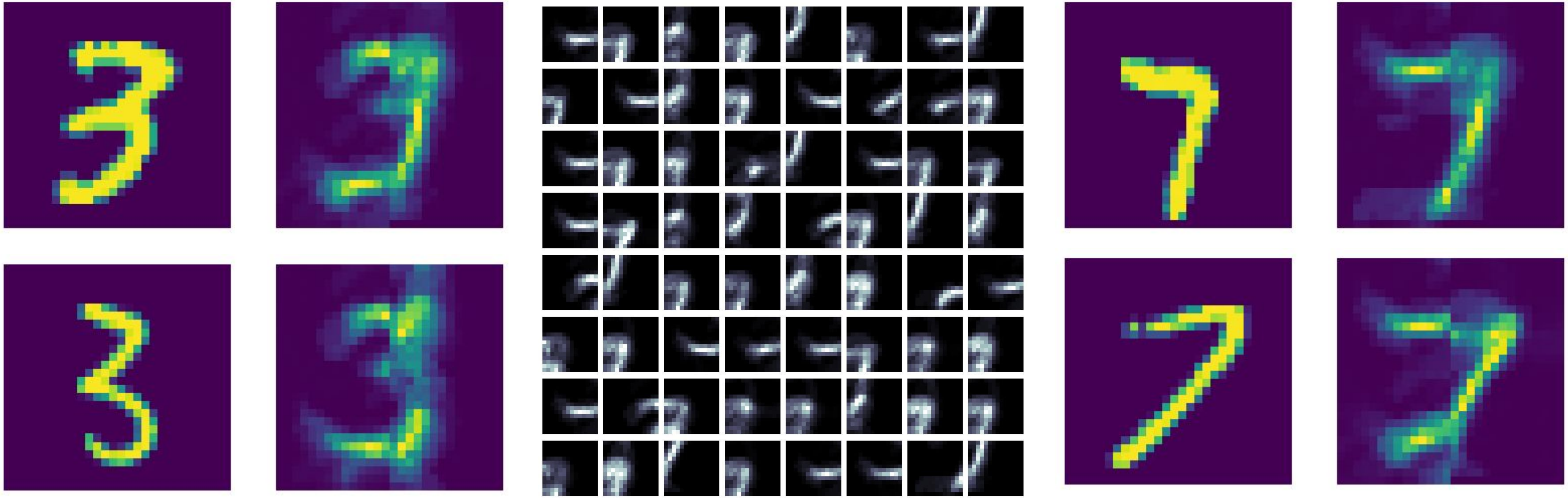
- What's more useful
  - The strokes of a three and a seven
  - The whole three and a seven
- Patches can be either overlapping or discrete
  - More general if overlapped, potentially to general

No Patches





With Patches



Seeing more data & diversity

- Seeing more data will always allow for a network to model the provided data better
- STDP tends to overfit to the data provided
- More diverse data slows this overfitting and allows it to generalize more

Where it is used

- Robotic Control
- Edge computing
- Computer Vision

Why it is used

- Lower power
- Naturally built for time dependent data
- Can be built on a hardware level

Questions?